

# Video

## ggplot2

ggplot, geom\_bar, xlab, scale\_y\_continuous, geom\_text

```
# video 1 國企二 吳予涵

rm(list = ls()) # Remove all the variables

packagesData <- data.frame(c("base", "dplyr", "ggplot2", "stringr", "readr"),
c(3, 13, 10, 6, 2))

plot1 <- ggplot(packagesData,
               aes(x= packagesData[,1],
                  y= packagesData[,2],
                  # Color of the plot depend on column 2
                  fill= packagesData[,2]))
plot1

# geom_bar for discrete bar charts, for continuous y, use geom_col()
# stat="identity" or stat="count"
plot2 <- plot1 + geom_bar(stat="identity", width = 0.2)
plot2

plot3 <- plot2 + xlab("Packages") +
  theme(legend.position = "none") +
  scale_y_continuous("Counts", breaks = c(0, 5, 10, 12), labels = c("0", "5",
"10", "12"))
plot3

plot4 <- plot3 + geom_text(aes(label = packagesData[,2]), data = packagesData)
plot4
```

geom\_blank, geom\_boxplot, geom\_smooth, geom\_facet\_grid, geom\_bar

```
# video 2 國企三 章君鈺

rm(list = ls()) # Remove all the variables

> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4          0.2  setosa
2           4.9         3.0          1.4          0.2  setosa
3           4.7         3.2          1.3          0.2  setosa
4           4.6         3.1          1.5          0.2  setosa
5           5.0         3.6          1.4          0.2  setosa

# The blank geom draws nothing, but can be a useful way of ensuring common scales
between different plots.
```

```

ggplot(iris,
      aes(x = Species, y = Sepal.Length)) +
  geom_blank()

# The boxplot compactly displays the distribution of a continuous variable.
# It visualises five summary statistics (the median, two hinges and two
whiskers), and all "outlying" points individually.
ggplot(iris,
      aes(x = Species, y = Sepal.Length)) +
  geom_boxplot()

# The point geom is used to create scatterplots.
ggplot(iris,
      aes(x = Sepal.Width, y = Sepal.Length)) +
  geom_point()

# Aids the eye in seeing patterns in the presence of overplotting.
ggplot(iris,
      aes(x = Sepal.Width, y = Sepal.Length)) +
  geom_point() +
  geom_smooth()

# facet_grid() forms a matrix of panels defined by row and column faceting
variables.
# It is most useful when you have two discrete variables, and all combinations of
the variables exist in the data.
# If you have only one variable with many levels, try facet_wrap().
ggplot(iris,
      aes(x = Sepal.Width,
          y = Sepal.Length)) +
  geom_point() + facet_grid(Species~.)

ggplot(iris,
      aes(x = Sepal.Width,
          y = Sepal.Length)) +
  geom_point() + facet_grid(.~Species)

ggplot(iris,
      aes(x = Petal.Width)) +
  geom_bar()

ggplot(iris,
      aes(x = Petal.Width)) +
  geom_bar(width = 0.01)

```

## ggplot, geom\_errorbar, geom\_rect, labs, xlim, ylim

```

# 國企三 毛子晴

rm(list = ls()) # Remove all the variables

load('datasets-V3.RData')

group <- c() # fall, rise, unchanged

```

```

for (i in 1:dim(data1)[1]){
  if (data1$Open[i]>data1$Close[i]){
    group[i]='fall'}
  else if (data1$Open[i]<data1$Close[i]){
    group[i]='rise'}
  else if (data1$Open[i]==data1$Close[i]){
    group[i]='unchanged'}
}

data2 <- cbind(data1, group)
data2$Date <- as.Date(data2$Date)

ggplot(data=data2, aes(x=Date)) + geom_errorbar(aes(ymin=Low, ymax=High),
size=0.3)

ggplot(data=data2, aes(x=Date)) +
  geom_errorbar(aes(ymin=Low, ymax=High), size=0.3) +
  geom_rect(aes(xmin=Date-0.3, xmax=Date+0.3,
                ymin=pmin(Open,Close), ymax=pmax(Open,Close), fill=group))

ggplot(data=data2, aes(x=Date)) +
  geom_errorbar(aes(ymin=Low, ymax=High), size=0.3) +
  geom_rect(aes(xmin=Date-0.3, xmax=Date+0.3,
                ymin=pmin(Open,Close), ymax=pmax(Open,Close), fill=group)) +
  labs(x='Date', y='Price', title='TSMC Historical Prices') +
  scale_fill_manual(values=c('Green', 'Red', 'Black'))

ggplot(data=data2, aes(x=Date)) +
  geom_errorbar(aes(ymin=Low, ymax=High), size=0.3)+
  geom_rect(aes(xmin=Date-0.3, xmax=Date+0.3,
                ymin=pmin(Open,Close), ymax=pmax(Open,Close), fill=group)) +
  labs(x='Date', y='Price', title='TSMC Historical Prices') +
  scale_fill_manual(values=c('Green', 'Red', 'Black')) +
  xlim(as.Date(c('2020-07-01', '2021-2-1')))) +
  ylim(c(400, 700))

```

---

Reference:

[ggplot2](#)

---